NPS52 -85-014

NAVAL POSTGRADUATE SCHOOL

Monterey, California

ABSOLUTE BOUNDS ON SET INTERSECTION AND UNION SIZES FROM DISTRIBUTION INFORMATION

Neil C. Rowe

September 1985

Approved for public release; distribution unlimited

Prepared for:

Chief of Naval Research
Arlington, VA 22217

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral R. H. Shumaker
Superintendent

D. A. Schrady
Provost

The work reported herein was supported in part by the Foundation Research Program of the Naval Postgraduate School with funds provided by the Chief of Naval Research.

This report was prepared by:

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS52-85-014	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ABSOLUTE BOUNDS ON SET INTERSECTION AND UNION SIZES FROM DISTRIBUTION INFORMATION		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Neil C. Rowe		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943-5100		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61152N; RR000-01-10 N0001484WR41001
11. CONTROLLING OFFICE NAME AND ADDRESS Chief of Naval Research Arlington, VA 22217		12. REPORT DATE September 1985 13. NUMBER OF PAGES 47
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

databases, query processing, statistical computing, statistical inequalities, sets, Boolean algebra, estimation.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Estimation of set intersection and union sizes is important for access method selection for a database. Absolute bounds on sizes are often much easier to compute than size estimates, requiring no distributional or independence assumptions, and can answer many of the same needs. We present a large compendium of quick closed-form bounds on set intersection and union sizes, each applying to a different situation; they can be expressed as rules, and managed by rule-based or "knowledge-base" architecture. These methods use general-purpose statistics precomputed on the data, and exploit homomorphisms

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(onto mappings) of the data items onto distributions that can be more easily analyzed. Our methods can be used anytime, but tend to work best when there are strong or complex correlations in the data. This circumstance is poorly addressed by the standard methods of independence-assumption and distributional-assumption estimates, and hence our methods fill a need.

Absolute bounds on set intersection and union sizes from distribution information

Neil C. Rowe

Department of Computer Science
Code 52
Naval Postgraduate School
Monterey, CA 93943

ABSTRACT

Estimation of set intersection and union sizes is important for access method selection for a database. Absolute bounds on sizes are often much easier to compute than size estimates, requiring no distributional or independence assumptions, and can answer many of the same needs. We present a large compendium of quick closed-form bounds on set intersection and union sizes, each applying to a different situation; they can be expressed as rules, and managed by rule-based or "knowledge-base" architecture. These methods use general-purpose statistics precomputed on the data, and exploit homomorphisms (onto mappings) of the data items onto distributions that can be more easily analyzed. Our methods can be used anytime, but tend to work best when there are strong or complex correlations in the data. This circumstance is poorly addressed by the standard methods of independence-assumption and distributional-assumption estimates, and hence our methods fill a need.

Classification:

Keywords: databases, query processing, statistical computing, statistical databases, inequalities, sets, Boolean algebra, estimation

1. Why bounds?

Good estimation of the sizes of set intersections is crucial to selection of efficient access methods for data in a database, especially when joins are involved. Such estimation is necessary for estimates of paging or blocks required. But often absolute bounds on the sizes of set intersections can serve the purpose of estimates, for several reasons:

1. Absolute bounds are more often possible to compute than estimates. Estimates generally require distributional assumptions about the data, assumptions that are sometimes difficult and awkward to verify, particularly for data subsets not much studied. Bounds require no assumptions.

2. Bounds are often easier to compute than estimates, because the mathematics, as we shall see, can be based on simple principles -- no integrals (possibly requiring numerical approximation) are needed as with distributions. This has long been recognized in computer science, as in the analysis of algorithms where worst-case (or bounds) analysis tends to be much easier than average-case.

3. Even when bounds tend to be weak, several different bounding methods may be tried and the best bound used. This paper gives some quite different methods that can be used on the same problems.

4. Bounds fill a gap in the applicability of set-size determination techniques. As we will discuss in the next section, good methods exist when one can assume independence of the attributes of a database, and some statistical techniques exist when on can assume strong but simple correlations between attributes. But until now there have been few techniques addressing situations with many and complicated correlations between attributes, as bounds can. Such situations tend to occur more with human-generated data than natural data, so with increasing computerization of routine bureaucratic activity we may see more of them.

5. Since choices among database access methods are usually discrete (yes-or-no), good bounds on the sizes of intersections can be just as helpful for making decisions as "reasonable-guess" estimates, when the bounds do not substantially overlap between alternatives.

6. Bounds in certain cases permit absolutely certain elimination (pruning) of possibilities, such as branch-and-bound algorithms and compilation of database access paths. As another example, bounds also help random sampling obtain a sample of fixed size from an unindexed set whose size is not known, since an error in retrieving too few items is much worse than retrieving too many (another pass through the entire set will be required to guard against page placement bias).

7. Bounds also provide an idea of the variance possible in an estimate, sometimes better than a difficult-to-obtain standard deviation. This is useful for evaluating retrieval methods, since a method with the same estimated cost as another, but tighter bounds, is usually preferable.

8. Sizes of set intersections are also queryable in their own right, particularly with "statistical databases" [15], databases designed primarily to support statistical analysis. If the users are doing "exploratory data analysis" [17], the early stages of statistical study of a data set, quick estimates are important and bounds may be sufficient. This was the basis of an entire statistical estimation system [13].

9. Bounds (and especially bounds on counts) are essential for analysis of security of

statistical databases from indirect inferences [5].

As with estimates, precomputed information is necessary to put bounds on set sizes. The more space allocated to precomputed information, the better bounds can be. Unlike most work with estimates, however, we will examine prior information besides counts, including extrema, frequency statistics, and fits to other distributions. We will emphasize upper bounds on intersection sizes, but we will also give some lower bounds, and also some bounds on set unions and complements. Since set intersections only make sense within a single relation we shall consider the data universe as a single relation.

Section 2 of this paper reviews previous research. Section 3 briefly mentions paging, and section 4 summarizes our method of obtaining bounds. Section 5 examines in detail the various frequency-distribution bounds, covering first upper bounds on intersections (section 5.1), lower bounds on intersections (section 5.2), bounds on unions (section 5.5), bounds on queries containing arbitrary Boolean expressions for sets (section 5.7), and concludes (section 5.8) with a summary of storage requirements for these methods. Section 6 evaluates these bounds both analytically and experimentally. Section 7 examines a different class of bounds. range-analysis. first for univariate ranges (section 7.1), then multivariate (section 7.2). Finally, section 8 discusses handling of updates to the database.

2. Previous work

Estimation of the sizes of intersections has been an important issue in querying performance for some time now, and a variety of work has addressed it. The emphasis has been almost entirely on developing estimates, not bounds. Various independence and uniformity assumptions have been suggested (e.g., [4] and [10]). These methods work well for data that has no or minor correlations between attributes and between sets intersected, and where bounds are not needed.

Christodoulakis [2] has estimated sizes of intersections and unions where correlations are well modelled probabilistically. He uses a multivariate probability distribution to represent the space of possible combinations of the attributes, each dimension corresponding to a set being intersected and the attribute defining it. The size of the intersection is then the number of points in a hyperrectangular region of the distribution. This approach works well for data that has a few simple but possibly strong correlations between attributes or between sets intersected, and where bounds are not needed. Its main disadvantages are that it requires extensive study of the data beforehand to estimate parameters of the multivariable distributions (and the distributions can change with time and later become invalid), and it only works for some databases, those without too many correlations between entities.

Similar work is that of [7]. They model the data by coefficients equivalent to moments. They do not use multivariate distributions explicitly, but use the independence assumption whenever they can. The rest of the time they partition the database along various attribute ranges (into what they call "betas", what [5] calls "1-sets", and what [11] calls "first-order sets") and model the univariate distributions on every attribute. This approach does allow modelling of arbitrary correlations in the data, both positive and negative, but requires potentially enormous space in its reduction of everything to univariate distributions. It can also be very wasteful of space, since it is hard to give different correlation phenomena different granularities of description. The method also only gives estimates, not bounds.

Some relevant work involving bounds on set sizes is that of [8], which springs from a quite different motivation that ours (handling of incomplete information in a database system), and does not entail anything beyond the simpler kinds of information that we discuss here (what we call levels 1 and 5). [9] investigates bounds on the sizes of partitions of a single numeric attribute using prior distribution information, but does not consider the much more important case of multiple attributes.

There has also been relevant work over the years on probabilistic inequalities [1]. We can divide counts by the size of the database to turn them into probabilities on a finite universe, and apply some of these mathematical results. However, the first and second objections of section 1 apply to this work: it usually makes detailed distributional assumptions, and is mathematically complex. For practical database situations we need something more general-purpose and simpler.

3. The relationship of set sizes to paging

Since set size estimates on a database are primarily used to estimate the required number of page accesses, we must relate set sizes to pages. For most databases and most user querying, query sets (especially set intersections) are small compared to the size of a database. Assuming a typical page size of 2000 bytes, if we can model the placement of records on pages as independent of their content, a common situation, the number of pages containing members of a query set is on the same order of magnitude as the size of the set. Thus the bounds on set sizes that we obtain in this paper are often close to bounds on the number of pages necessary to retrieve. Under any circumstance the number of pages is always less than or equal to the number of items, so an upper bound on the set size (what most of the results in this paper represent) is an upper bound on the number of pages. But the lower bound is not so guaranteed.

4. The general method

We present two main approaches to calculation of absolute bounds on intersection sizes in this paper, frequency-distribution (section 5) and range-analysis (section 7). They are instances of a general method.

4.1. Two motivating examples

The two approaches occur in many guises, some quite simple. Suppose we have a census database on which we have tabulated statistics of state, age, and income. Suppose we wish an upper bound on the number residents of Iowa that are between the ages of 30 and 34 inclusive, when all we know are statistics on Iowa residents and statistics on people age 30-34 separately. One upper bound would be the frequency of the mode (most common) state for people age 30-34. Another would be five times the frequency of the most common age for people living in Iowa (since there are five ages in the range 30-34). These are examples of frequency-distribution bounds, to which we devote primary attention in this paper.

Suppose we also have income information in our database, and suppose the question is to find the number of Iowans who earned over 100,000 dollars last year. Even though the question has nothing to do with ages, we may be able to use age data to answer this question. We obtain the maximum and minimum statistics on the age attribute of the set of people who earned over 100,000 dollars (combining several subranges of earnings to get this if necessary), and then find out the number of people in the universe that lie in that age range, and that is an upper bound. We can also use the methods of the preceding paragraph to find the number of Iowans lying in that age range. This is an example of a range-restriction bound.

4.2. The method

Our basic method is quite simple. Before any queries are issued, preprocess the data:

(1) Group the data items into categories. The categories may be arbitrary.

(2) Count (aggregate) the number of items in each category, separately for each set being intersected.

Now when bounds on a set intersection or union are needed:

(3) Look up counts relevant to all the sets mentioned in the query.

(4) Find the minima (for intersections) or maxima (for unions) of the corresponding counts for each set in the query.

(5) Sum up the minima (or maxima) to get a bound on the intersection size.

The first main approach we take is grouping by data values for a single symbolic (numeric or nonnumeric) attribute (section 5); the second approach is grouping by subrange of a numeric attribute (section 7). The methods can be extended to lower bounds (section 5.2) and set unions (section 5.5).

4.3. General comments

All our rules for bounds on sizes of set intersections will be expressed as hierarchy of different "levels" of statistics knowledge about the data. Lower levels mean less prior knowledge, but generally poorer bounding performance.

The word "value" may be interpreted as any equivalence class of data attribute values. This means that prior counts on different equivalence classes may be used to get different bounds on the same intersection size, and the best one taken, though we do not include this explicitly in our formulae.

5. Frequency-distribution bounds

We now examine bounds derived from knowledge (partial or complete) of frequency distributions of attributes.

5.1. Upper frequency-distribution bounds

5.1.1. Level 1: set sizes of intersected sets only

Suppose we know the sizes of the sets being intersected. Then an upper bound on the size of the intersection is the minimum of the set sizes, or

$$\min_{i=1}^{s} n(i)$$

where $n(i)$ is the size of the ith set and s is the number of sets. And if we only know upper bounds on the sizes of some or all the intersected sets, an upper bound on the intersection is the minimum of the upper bounds.

5.1.2. Level 2a: mode frequencies and numbers of distinct items

Suppose for some attribute A of the sets that we know the mode frequency and number of distinct values for each set. Then an upper bound on the size of the intersection is the product of the minimum of the mode frequencies of each set on the attribute, with the minimum of the number of distinct items for each set with respect to the attribute, or:

$$(\min_{i=1}^{s} m(i,j)) * (\min_{i=1}^{s} d(i,j))$$

To prove this: (1) an upper bound on the mode frequency of the intersection is the minimum of the mode frequencies; (2) an upper bound on the number of distinct items of the intersection is the minimum of the number for each set; (3) an upper bound on the size of a set is the product of its mode frequency and number of distinct values; and (4) an upper bound on the product of two nonnegative uncertain quantities with upper bounds is the product of their upper bounds.

As an example, suppose we have sets of sizes 1000, 2000, and 1500 with corresponding mode frequencies on some attribute of 300, 200, and 100, and with corresponding numbers of distinct values 4, 50, and 30. Then the bound on the size of the intersection is 100 * 4 = 400.

As before, the rule still applies if we know only upper bounds on the mode frequencies and numbers of distinct values for the sets. In particular, note the size of a set is always an upper bound on both, and this can be useful.

If we know information about more than one attribute of the data, we can simply take the minimum of the upper bound computations on each attribute. To put this generalization formally, let m(i,j) be the mode frequency of set i with respect to attribute j, and let d(i,j) be the number of distinct items. Let s be the number of sets being intersected, and r the number of attributes we know these statistics about (not necessarily all the attributes in the database). Then the bound is

$$\min_{j=1}^{r}\left[(\min_{i=1}^{s} m(i,j)) * (\min_{i=1}^{s} d(i,j))\right]$$

5.1.3. Some important special cases

Two important corollaries follow immediately from the preceding result. These were used in our previous work [13] until we discovered the generalization.

A special case occurs when one set being intersected has only one possible value on a given attribute -- that is, the number of distinct values is 1. This condition can arise when a set is defined as a partition of the values on that attribute, but also can occur accidentally, particularly when the set concerned is small. Hence the second inner minima in the above formal expression is 1, and the value of the whole expression is the first of the inner minima, or the minimum of the mode frequencies on that attribute. For example, an upper bound on the number of American tankers is the mode frequency of tankers with respect to the nationality attribute.

The second special case is the other extreme, when one set being intersected has all different values for some attribute, or a mode frequency of 1. This arises from what we call an "extensional key" ([11], ch. 3) situation, where some attribute functions like a key to a relation but only in a particular database state. Hence the first inner minimum in the above formal expression is 1, and the value of the whole expression is the second inner minimum, or the minimum of the number of distinct values on that attribute. For example, an upper bound on the number of American tankers in Naples, when we happen to know the port has only one ship of a given nationality at a time, is the number of nationalities of tankers in Naples.

5.1.4. Level 2b: a different bound with the same information

A different line of reasoning leads to a different bound utilizing mode frequency and number of distinct items, giving an "additive" bound instead of the "multiplicative" one above. This bound has an important advantage over the level 2a bound: it is always less than the level 1 bound, which is not necessarily true for 2a. But in other situations 2a can be better than this new bound, which we call 2b.

Consider the mode on some attribute as partitioning a set into two pieces, those items having the mode value of the attribute, and those not. Then a bound on the the size of the intersection of r sets is

$$\min_{j=1}^{r}\left[\min_{i=1}^{r} m(i,j)+\min_{i=1}^{r}\Big(n(i)-m(i,j)\Big)\right]$$

To prove this, denote the items in set i having the mode value on the attribute j as M_i, and denote the rest of the set i as R_i. Then an upper bound on the mode frequency of the intersection is the minima of the sizes of the M_i, because even if the M_i do not correspond to the same value, interchanges with values within the R_i that would make the M_i all correspond to the same value could not increase the sizes of the M_i, by the definition of the mode. Similarly, an upper bound on the size of the rest of the intersection set is the minima of the sizes of the R_i, since if we let k be the index of the set with smallest-size R_i, the interchanges between mode and any item within the R_i for any other set than k, necessary to get every value in R_k matched to its counterparts in R_i, cannot but increase the size of R_i -- and since R_i is by assumption bigger than R_k, this cannot increase the minimum of the R_i sizes.

As an example, take the case we considered before where we have sets of sizes when 1000, 2000 and 1500 with corresponding mode frequencies of 300, 200 and 100. Then the formula is

$$\min(300,200,100)+\min(1000 \quad 300,2000 \quad 200,1500-100)=800$$

So in this case the level 2b bound is worse than the level 2a bound. But this is not true for all cases. If we are intersecting two sets of size 1000, where the mode frequency of the first set is 100 and the mode frequency of the second is 900, and the number of distinct values for both sets is 20, then the level 2a bound is 2000 and the level 2b bound is 200. So the level 2b bound is better in this case.

But the above bound doesn't use the information about the number of distinct values. If the set i that minimizes the last minima in the formula above contains more than the minimum of the number of distinct values d(i,j) over all the sets, we must "subtract out" the excess, assuming conservatively that the extra items occur only once in set i:

$$\min_{j=1}^{r}\left[\min_{i=1}^{s} m(i,j)+\min_{i=1}^{s}(n(i)-m(i,j)+\min_{k=1}^{s}(d(k,j)-d(i,j)))\right]$$

The above makes the conservative assumption that values not in the set i occur only once in the other set k. It would seem that we could do better by subtracting out the minimum mode frequency from set i over the sets a number of times corresponding to the minima of the the number of distinct values over all the sets. However, it turns out this reduces to the level 2a bound.

5.1.5. Level 2c: Diophantine inferences from sums

A very different kind of information about a distribution than its mode frequency and number of distinct values is sometimes useful when the attribute is numeric: its sum and other moments on the attribute for the set. (Since the sum and standard deviation require the same amount of storage as level 2a and 2b information, we classify them as another level 2 situation.) This information is only useful when (a) we know the set of all possible values for the universal set. and (b) there are few of these values relative to the size of the sets being intersected. Then we can write a linear Diophantine (integer-solution) equation in unknowns representing the number of items of each particular numeric value in each of the sets being intersected, and each solution represents a possible partition of counts on each value. An upper bound on the intersection size is thus the sum over all values of the minimum over all sets of the maximum number of occurrences of a particular value for a particular set. See [12] for a further discussion of Diophantine inferences about statistics. A noteworthy feature of Diophantine equations is the unpredictability of their power: often there are many solutions, but sometimes for no apparent reason there are only a few, and inferences are powerful.

5.1.6. Level 3a: other piecemeal frequency distribution information

The level 2 approach will not work well for sets and attributes that have large mode frequencies compared to the other set frequencies.. We could get a better (i.e. lower) upper bound if we knew the frequencies of other items than the mode, like the second most common item and the median frequency item. For the first, we can subtract out the mode frequency first. giving for m2(i,j) representing the frequency of the second most common item of the ith set on the jth attribute:

$$\min_{j=1}^{r}\left[(\min_{i=1}^{s} m(i,j))+(\min_{i=1}^{s} m2(i,j))*((\min_{i=1}^{s} d(i,j))-1)\right]$$

To justify this we must prove that the frequency of the second most common item of the intersection cannot occur more than the minimum of the second most common items of those sets. Prove this by contradiction. Let M be the mode frequency of the intersection and let M2 be the frequency of the second most common item in the intersection. Assume M2 is more than the frequency of the second most common item in some set i. Then M2 must correspond to the mode of that set i. But then the mode of the intersection must be less than or equal to the mode of the second most frequent item in set i, which is a contradiction.

Using the same example we have used previously, assume we want to intersect three sets of sizes 1000, 2000 and 1500, where the mode frequencies are 300, 200, and 100 respectively, and where the number of distinct values for each set are 4, 50, and 30. Now assume the second most frequent item has frequency 250, 150, and 60 for those three sets respectively. Then the formula gives us

$$\min(300,200,100)+\min(250,150,60)*(\min(4,50,30)-1)=280$$

For knowledge of the frequency of the median-frequency item (call it mf(i,j)), we can just divide the outer minimum into two parts:

$$\min_{j=1}^{r}\left[(\min_{i=1}^{s} m(i,j))*(\min_{j=1}^{s} d(i,j))+(\min_{i-1}^{s} mf(i,j))*(\min_{j=1}^{s} d(i,j))\right]$$

Knowledge of the mean frequency is no use to us since this is just the set size divided by the number of distinct items.

5.1.7. Level 3b: a different bound using the same information

In the same way that level 2b complements level 2a, there is 3b bound that complements the preceding 3a bound. Following analogous reasoning to the discussion of 2b, an upper bound on the intersection size is:

$$\min_{j=1}^{r}\left[\min_{i=1}^{s} m(i,j)+\min_{i=1}^{s} m2(i,j)+\min_{i=1}^{s}\left[n(i)-m(i,j)-m2(i,j)\right.\right.$$
$$\left.\left.+\min_{k=1}^{s}(d(k,j)-d(i,j))+\min_{k=1}^{s}(mf(k,j)-mf(i,j))\right)\right]\right]$$

If the median frequency is unknown, drop the last minimum. The formula can be improved still further if we know the frequency of the least common item on set i, and it is greater than 1: just multiply (d(k,j)-d(i,j)) by this least frequency.

As an example take the same numbers as in the previous section, and the bound is

$$100+40+\min(1000-300-250-0.2000-200-150-46,1500-100-60-26)=590$$

5.1.8. Level 4a: full frequency distribution information

An obvious extension is to situations where we know the frequency distribution (histogram) for an attribute for each set, but not which value has which frequency. This sounds like a good deal of information, but may not be for few-valued attributes.

Now we do not need to use the number of distinct values since this is given implicitly. By similar reasoning to that we gave in the last section for the frequency of the second most common item, we can give the bound as:

$$\min_{j=1}^{r} \sum_{k=1}^{d(U,j)} \min_{i=1}^{s} freq(i,j,k)$$

where freq(i,j,k) is the frequency of the kth most frequent value of the ith set on the jth attribute. To prove this, assume the kth most common item in the intersection occurs some f number of times which is more than the minimum of the kth most common items in each of the sets. Then in some set i there must be an item that occurs f times, and this item must be ranked less than k in order for f to be greater than the minimum of all the kth-best items. But then since the first k items are all distinct, one of the kth-best items in the intersection must occur at greater than a rank of k in set i. But then the minimum over all the sets for this ranking would have to be at most this number less than f, and hence a higher-ranked item would occur in the intersection less often than a lower-ranked item, a contradiction.

As an example, suppose we want to intersect three sets whose distributions are (50, 14, 10, 6, 1), (30, 15, 7, 2, 0), and (22, 17, 12, 9, 2) as sorted by decreasing frequency. Then the level 4 bound is $22+14+7+2+0=45$.

Note we can still use this formula if all we know is an upper bound on the actual distribution -- we just get a weaker bound. Thus there are many gradations between level 3 and level 4. This is useful because if one can find a classical probability distribution (like a normal curve) that lies entirely above the curve, it can be specified with just a few parameters, thus saving a good deal of space over exact specification of an entire distribution.

As an example, suppose we have two exponential distributions on the range between 0 and 2. Suppose we can upper-bound the first distribution by $100e^{-x}$ and the second by $100e^{x-2}$, so there are about 86 of each set. Then the distribution of the intersection is bounded above by the minimum of those two distributions. So an upper bound on the size of the intersection is

$$\int_0^1 (100e^{x-2})\,dx + \int_1^2 (100e^{-x})\,dx = 100(e^{-1} - e^{-2} - e^{-2} + e^{-1}) = 46.6$$

5.1.9. Level 4b: Diophantine inferences about values

A different kind of Diophantine inference than that discussed in 5.1.5 can arise when the data distribution is known for some numeric attribute: we may be able to use the sum (plus other moments) to infer possible values for each set being intersected, and use this to bound the number of possible values in the intersection. To make Diophantine solution practical we require that (a) the number of distinct values in each set being intersected is small with respect to the size of the set, and (b) the least common divisor of the possible values be not too small (say less than .001) of the size of the largest possible value. Then we can write a linear Diophantine equation in unknowns which this time are the possible values, and solve for all possibilities. Again, see [12] for further details.

5.1.10. Level 5: tagged frequency distributions

Finally, the best kind of frequency distribution information we could have about sets would be not only the distribution for each, but which values in each distribution match up with which other values in other distributions. This gives an upper bound of:

$$\min_{j=1}^{r} \sum_{k=1}^{d(U,j)} \min_{i=1}^{s} gfreq(i,j,k)$$

where gfreq(i,j,k) is the frequency of globally-numbered value k of attribute j for set i, which is zero when value k does not occur in set i, and where d(U,j) is the number of distinct values for attribute j in the universe.

As an example, suppose we want to intersect the same three sets in the example in section 5.1.8, whose distributions on some same order of attributes of (10, 6, 50, 14, 1), (30, 2, 7, 15, 0), and (12, 9, 17, 22, 2). Then the level 5 upper bound is 10+2+7+14+0=33, which is better than the level 4 upper bound we got above on the same data.

All that is necessary to identify values is a unique code, not necessarily the actual value. Bit strings can be used together with an (unsorted) frequency distribution of the values that do occur at least once. Again note that an upper bound on the actual distribution can be used instead to give a weaker (but easier to specify) bound.

Notice that level 5 information is analogous to level 1 information, as it represents sizes of particular subsets formed by intersecting each original set with the set of all items in the database having a particular value for a particular attribute. This is what [11] calls "second-order sets" and [5] "2-sets". Thus we have come full circle, and there can be no "higher" levels than 5.

5.2. Lower bounds from frequency distributions

On occasion we can get nonzero lower bounds on the size of a set intersection. These situations commonly arise when the size of the "universe" set (the set of all items in the database) is known, and the sets being intersected are almost the size of the universe.

5.2.1. Lower bounds: levels 1 and 5

By Boolean algebra we know that the intersection of some sets is the same as the complement (with respect the universe) of the union of the complements of each set. An upper bound on the union of some sets is the sum of their set sizes. An upper bound on the union of s sets of sizes n(i) with a universe of size N is

$$\sum_{i=1}^{s} (N - n(i))$$

and hence a lower bound on the size of the intersection of those sets is

$$\max(0, N - \sum_{i=1}^{s} (N - n(i))) = \max(0, \sum_{i=1}^{s} n(i) - (s - 1)N)$$

which is the statistical form of the simplest case of the Bonferroni inequality. For most sets of interest to a database user this will be less than zero and hence useless, since note that sN is the largest possible value for any sum. But with only two sets being intersected. or sets corresponding to outlier-removing restrictions, a nonzero lower bound may more often occur.

As an example, suppose we have a universe of size 1000, and we want to intersect two sets of size 700 and 800. Then the lower bound is $700+800-1000=500$.

We can use this idea for level 5 information where we know the exact frequency distribution of the values and can match values between distributions being intersected. Then the above expression applies separately to each possible value, and we have for a lower bound

$$\max_{j=1}^{r} \sum_{k=1}^{d(U,j)} \max(0, \left(\sum_{i=1}^{s} gfreq(i,j,k) \right) - (s-1) gfreq(U,j,k))$$

where gfreq(i,j,k) is as before the number of occurrences of the kth most common value of the jth attribute for the ith set, and d(U,j) is the number of distinct values for attribute j in the universe.

As an example, suppose we want to intersect two sets withs distributions (13, 25, 8, 4, 1) and (12, 2, 1, 5, 2), where the universe has distribution on these same attributes in order of (20, 40, 8, 7, 3). Then the level 5 lower bound is $5+0+1+2+0=8$..

5.2.2. Lower bounds: levels 2, 3, and 4

It is more difficult to obtain nonzero lower bounds when information is not tagged to specific sets, as what we have called levels 2, 3, and 4. For instance, the mode frequency and number of distinct values of the intersection can almost always be zero (i.e., the sets can have an empty intersection).

Occasionally, if we know the mode values as well as the mode frequencies, and the modes are all identical, we can match mode frequencies as in the last equation above, and bound the frequency of the mode in the intersection (this must be nonzero if modes are identical). Then if we know a lower bound on the number of distinct values in the intersection, we multiply the latter by the former to get a set size lower bound. Even if we do not know the modes, we may be able to infer that the items having the mode frequencies are identical if we know instead the frequencies of the second most common items. and these are sufficiently far from the mode frequency for every set such that

$$m(i,j) - m2(i,j) > N - n(i)$$

where m(i,j) is the mode frequency of set i on attribute j, m2(i,j) the frequency of the second most common item, N the size of the universe, and n(i) the size of the ith set.

The problem for level 4 lower bounds is that we do not know which items in the frequency lists correspond to which values. But if we have some computer time to spend, we can exhaustively consider all the combinatorial possibilities, excluding those impossible given the frequency distribution of the universe, and take as the lower bound the level-5 bound for the situation with the lowest one. For instance, with an implementation of this method in Prolog (a language well suited for this kind of problem), we considered a universe with four data values, where the frequency distribution of the universe was (54, 53, 52, 51), and the frequency distributions of the two sets intersected were (40, 38, 22, 20) and (30, 23, 21, 16). The level 4 lower bound was 8, and occurred for several matchings, including the one:

$$(54-38 \quad 21.53-40-16,52-22-30.51-20-23)$$

For comparision, the level 1 lower bound is 210 - 120 - 90 = 0, so the effort may be worth it. (Note also the level 1 and 4 upper bounds are both 30 + 23 + 21 + 16 =

90.) But the number of combinations that must be considered for k distinct values in the universe is $(k!)^2$ -- which is 14,400 for k=5, so this idea will only work for very few numbers of distinct items.

5.2.3. Definitional sets

There is another, very different way of getting lower bounds, from knowledge of how the sets intersected were defined. If we know that set A was defined as all items having particular values for an attribute j, then in analyzing an intersection including set A, the "definitional" set A contributes no restrictions on attributes other than j and can be ignored. This is redundant information with levels 1 and 5, but it may help with the other levels. For instance, if set A is definitional on j, a lower bound on the size of the intersection of sets A and B is the frequency of the least frequent item (the "antimode") of set B on j.

5.3. Better bounds from relaxation on sibling sets

Both upper and lower bounds can possibly be improved by relaxation on sibling sets (those having sets in common in the intersection lists defining them), in the manner [3], work aimed at protection of data from statistical disclosure. This approach requires a good deal more computation time than of the closed-form formulae in this paper and requires sophisticated algorithms. Thus we do not discuss it here.

5.4. Incomplete knowledge of sets queried

It is straightforward to extend analysis to partial but bounded knowledge of the frequency statistics (or sets sizes) on some or all the sets being analyzed. For upper bound rules, we can just use an upper bound on the frequency statistics; for lower bound rules, the lower bound on the frequency statistics.

5.5. Set unions

So far we have examined only set intersections. Intersections are usually more common than unions in user querying of a database, since they have many more applications to real-world problems. But rules analogous to intersection rules for unions and complements are not hard to obtain.

5.5.1. Defining unions from intersections

Suppose we want to bound the union of two sets i and j. The size of the union of two sets i and j is

$$n(i \bigcup j) = n(i) + n(j) - n(i \bigcap j)$$

where $n(i \bigcup j)$ means the size of the union of set i and set j, and $n(i \bigcap j)$ means the size of their intersection, extending our previous notation for set size. Hence

$$n(i \bigcup j \bigcup k) = n(i) + n(j) + n(k) - n(i \bigcap j) - n((i \bigcup j) \bigcap k)$$

$$= n(i) + n(j) + n(k) - n(i \bigcap j) - n(i \bigcap k) - n(j \bigcap k) + n(i \bigcap j \bigcap k)$$

using the distribution of intersection over union, and these results can be extended in a well-known way to the union of arbitrary numbers of sets. In general:

$$n\left(\bigcap_{i=1}^{s} A(i)\right) = \sum_{i=1}^{s} n(i) - \sum_{i1=1}^{s} \sum_{i2=1, i2 \neq i1}^{s} n(i1 \cap i2)$$

$$+ \sum_{i1=1}^{s} \sum_{i2=1, i2 \neq i1}^{s} \sum_{i3=1, i3 \neq i2, i3 \neq i1}^{s} n(i1 \cap i2 \cap i3) - \cdots$$

Thus all set expressions involving unions can be reduced to simple arithmetic operations on expressions involving intersections only. So statistics on unions reduce to statistics on intersections.

Another approach to unions is to use complements of sets and DeMorgan's Law:

$$\overline{\bigcup_{i=1}^{s} A(i)} = \bigcap_{i=1}^{s} \overline{A}(i)$$

$$n\left(\bigcup_{i=1}^{s} A(i)\right) = N - n\left(\bigcap_{i=1}^{s} \overline{A}(i)\right)$$

The problem with using this is the computing of statistics on the complement of a set. This is easy for counts, but difficult for mode frequency, number of distinct values, and the other level 2, 3, and 4 information.

In one important situation the calculation of union sizes is particularly easy: when the two sets unioned are disjoint (that is, their intersection is empty). Then the size of the union is just the sum of the set sizes, by the first formula in this section. Disjointness can be known a priori, or we can infer it using methods to be discussed in section 7.1.2.

5.5.2. Level 1 information for unions

To obtain union rules from intersection rules, we can do a "compilation" of the above formulae (see section 3.5.5 of [11] for other examples of this process) by substituting rules for intersections in them, and simplifying the result. As an example, substitute the level 1 intersection bounds in the above set-complement formula:

$$inf\left(n\left(\bigcup_{i=1}^{s} A(i)\right)\right) = N - \left(\min_{i=1}^{s}(N - n(i))\right) = N + \max_{i=1}^{s}(n(i) - N)$$

$$= \max_{i=1}^{s} n(i)$$

$$sup\left(n\left(\bigcup_{i=1}^{s} A(i)\right)\right) = N - \max\left(0, \left(\sum_{i=1}^{s}(N - n(i))\right) - (s-1)N\right)$$

$$= N - \max\left(0, N - \sum_{i=1}^{s} n(i)\right) = N + \min\left(0, -N + \sum_{i=1}^{s} n(i)\right)$$

$$= \min\left(N, \sum_{i=1}^{s} n(i)\right)$$

The lower bound occurs when the sets are all disjoint; the upper bound occurs when some set contains all the others.

5.5.3. Level 2b unions

We can obtain lower bounds on union sizes with level 2, 3 and 4 information analogously to the way we obtained upper bound on intersection sizes with that information. We start with level 2b, since it is the easiest.

If all we know is the mode frequency m(i,j) and the number of distinct values of attributes d(i,j): (1) we know a lower bound on the mode frequency of the union is the maxima of the mode frequencies; (1) a lower bound on the number of distinct values is the maxima of the number of distinct values on each attribute. Since we know sizes of two disjoint subsets for each set -- the number of items have the mode value, and the number of non-mode items -- we can use a formula analogous to the level 2b intersection upper bound:

$$\max_{j=1}^{r}\left(\left(\max_{i=1}^{s} m(i,j)\right)+\left(\max_{i=1}^{s}(n(i)-m(i,j))\right)\right)$$

Taking number of distinct values into account gives a formula analogous to the other level 2b intersection bound: If the set i that fulfills the second maximum does not contain the maximum number of distinct values among the sets, then the missing values must be included in the count too, giving:

$$\max_{j=1}^{r}\left[\max_{i=1}^{s} m(i,j)\right.$$
$$\left.+\max_{i=1}^{s}(n(i)-m(i,j)+\max_{k=1}^{s}(d(k,j)-d(i,j)))\right]$$

5.5.4. Level 2a unions

The approach used in level 2a for intersections is difficult to use here. We cannot use the negation formula to relate unions to intersections because there is no comparable multiplication of two quantities (like mode frequency and number of distinct values) that gives a lower bound on something. However, we can use the other (first) formula relating unions to intersections. Since a lower bound on the size of the intersection of s sets is a lower bound on the size of any two of them, we can use:

$$inf(n(A(i1)\bigcup A(i2)))=n(i1)+n(i2)-\min(m(i1,j),m(i2,j))*\min(d(i1,j),d(i2,j))$$

Or if we have three sets we want to intersect, we can use the following:

$$n(i1)+n(i2)+n(i3)$$
$$-\min_{j=1}^{r}\left[\min(m(i1,j),m(i2,j))*\min(d(i1,j),d(i2,j))\right.$$
$$-\min(m(i1,j),m(i3,j))*\min(d(i1,j),d(i3,j))$$
$$\left.+\min(m(i2,j),m(i3,j))*\min(d(i2,j),d(i3.j))\right]$$
$$+\max_{j=1}^{r}\left[\max(m(i1,j).m(i2,j),m(i3,j))*\max(d(i1,j),d(i2,j),d(i3,j))\right]$$

5.5.5. Level 3b unions

Analogous to level 2b, we

$$\max_{j=1}^{r}\left[\max_{i=1}^{s} m(i,j)+\max_{i=1}^{s} m2(i,j)\right.$$

$$+\max_{i=1}^{s}(n(i)-m(i,j)-m2(i,j)+\max_{k=1}^{s}(d(i,j)-d(k,j))+\max_{k=1}^{s}(mf(i,j)-mf(k,j)))]$$

where m2 is the frequency of the second most common item, and mf the frequency of the median-frequency item. And if we know the frequency of the least common item in set i, we multiply (d(i,j)-d(k,j)) by it.

5.5.6. Level 3a unions

Analogous to level 2a, and to level 3a intersections, we have for two sets only a lower bound of:

$$n(i1)+n(i2)-\sum_{j=1}^{r}\Big(\min(d(i1,j),d(i2,j))$$

$$*\,[\min(m(i1,j),m(i2,j))+\min(mf(i1,j),mf(i2,j)))]$$

where m2 is the frequency of the second most common item, and mf the frequency of the median-frequency item.

5.5.7. Level 4 unions

Analysis of level 4 is analogous to that of section 5.1.6, giving a lower bound of

$$\max_{j=1}^{r}\left(\sum_{k=1}^{d(U,j)}\max_{i=1}^{s} freq(i,j,k)\right)$$

where freq(i,j,k) is the frequency of the kth most frequent value of the ith set unioned on the jth attribute. So we have just replaced the minima by maxima. This formula can be proved by the analogous argument to that given in 5.1.6.

5.5.8. Level 5 unions

Level 5 is analogous to level 1:

$$inf\colon \max_{j=1}^{r}\sum_{k=1}^{d(U,j)}\max_{i=1}^{s} gfreq(i,j,k)$$

$$sup\colon \min_{j=1}^{r}\sum_{k=1}^{d(U,j)}\min(\left(\sum_{i=1}^{s} gfreq(i,j,k)\right),gfreq(U,j,k))$$

Note for the upper bound, if the second argument to the inner min is never the minimum, the level 5 formula becomes identical to the level 1 formula.

5.6. Complements

To complete our coverage of set algebra we need set complements. But counts on them are easy for sets that are not composite. The size of its complement is the difference of the size of the universe (something that is often important, so we undoubtedly will know it) and the size of the set. If the set is composite (composed of intersections and unions of other sets) things are more complicated, and we discuss this approach in the next section.

5.7. Embedded query expressions

So far we have only considered intersections, unions, and complements of simple sets about which we know exact statistics). But if the query language permits arbitrary embedding of query expressions (and many do), new complexities. First, we must use the methods of section 5.4 and find upper bounds and long bounds on statistics for each composite term, substituting upper bounds in maxima and lower in minima. But also, there can be many equivalent forms of a Boolean algebra expression, and we have to careful which equivalent form we choose, because different forms give different bounds.

5.7.1. Summary of equivalences

The Appendix surveys the effect of various equivalences of Boolean algebra on bounds using level 1 information. It turns out that commutativity and associativity do not affect bounds, but factoring out of common sets in conjuncts or disjuncts with distributive laws is important, since it usually gives better bounds. Factoring out enables other simplification laws which usually give better bounds too. So factoring of either disjuncts or conjuncts seems very important to handling of embedded queries with level 1 information.

The formal summary of the Appendix is as follows ("yes" means in all but trivial cases):

Equivalence	Better sup?	Better inf?
Commutativity	no	no
Reflexivity	yes	yes
Associativity	no	no
Factoring of $\bigcap$ over $\bigcup$	yes	no
Factoring of $\bigcup$ over $\bigcap$	no	yes
Operations with U and Φ	no	no
Absorption	yes	yes
Identity elements	yes	yes
Negation-absorption	yes	yes
DeMorgan's Laws	no	no

Since these equivalence transformations are sufficient to derive any equivalent query expression from a query expression, the entries in the above table are the only information necessary to decide whether one query form is better than another for all possible sets.

5.7.2. The "best" form of a given query, for level 1 information

So the best query form for the best level 1 bounds is a highly factored form, just the opposite of a disjunctive normal form or a conjunctive normal form. The number of Boolean operators doesn't matter, more the number of sets they operate on, so we don't want "minimum-gate" form derived from Karnaugh maps. So minimum-term form [6] seems to be closest to what we want. Note that all the useful transformations in the above table reduce the number of terms in an expression, so term minimization seems a good heuristic (though this is not a proof). It makes sense because more than one occurrence of the same set in a query should be expected to cause suboptimal bounds -- the bounds calculations all assume the independence of the sets mentioned in the query, or that the sets are all different.

But note these three equivalent expressions

$$(A \cap (B \cup C)) \cup (B \cap C) = (B \cap (A \cup C)) \cup (A \cap C) = (C \cap (A \cup B)) \cup (A \cap B)$$

cannot be ranked with respect to one another, though they are all preferable to the unfactored equivalent

$$(A \cap B) \cup (A \cap C) \cup (B \cap C)$$

So we must accept multiple "best" forms for a query. We can compute the bounds on each form, and intersect the ranges to get the cumulative range. This should not arise very often, because users will tend to issue queries with few repeated mentions of the same set -- parity queries are rarely needed. So the following heuristic method will usually obtain the "best" query form:

1. Write the query in disjunctive normal form, eliminating duplicates in each conjunction.

2. Choose the set that occurs in the most terms (counting the complement of a set as a different set), and factor it out.

3. Apply any absorption and term elimination laws possible to what is left after factoring.

4. Return to step 2 and look for more things to factor out. Continue until no more factorizations are possible.

5. Apply the same idea (with the dual factorization and absorption laws) starting with the conjunctive normal form.

6. Return the fully factored form of the two -- the factored disjunctive normal form, and the factored conjunctive normal form -- that had fewer terms.

5.7.3. Embedded queries with other levels of information

Level 5 is analogous to level 1 -- it just represents a partition of all the sets being intersected into subsets of a particular range of values on a particular attribute, with bounds being summed up on all such ranges of values of the attribute. Thus the above "best" query forms will be equally good for level 5 information. For level 1 analysis considerably more complicated for levels 2, 3, and 4 since we do not have both upper and lower bounds in those cases. But the "best" forms can be used heuristically.

5.8. Analysis of storage requirements

The different levels of prior information require different amounts of storage. We will give some rough storage requirements here, to enable comparison to the performance analysis in section 6.

5.8.1. Some formulae

Assume a database consisting of a single relation (file or table) of r attributes on N items, each attribute value requiring w bits of storage. The database thus requires rNw bits of storage. Assume we only tabulate statistics on "1-sets" [5] or "first-order sets" [11] or partitions of the items according to values on a single attribute. Assume there are m approximately even partitions on each attribute. Then the space required

for storage of statistics is as follows:

> Level 1: there are mr sets with just a set size tabulated for each. Each set size recorded requires about $\log_2(N / m)$ bits, so $mr^* \log_2(N / m)$ total bits are required. This will tend to be considerably less than rNw, the size of the database, because w will likely be on the same order as $\log_2(N / m)$, and m is considerably less than N.
>
> Level 2: for each of the mr sets we have 2r statistics (the mode frequency and number of distinct values for each attribute). (This assumes we do not have any criteria to claim certain attributes as being valueless, as when attributes exhibit no significantly different distributions for different sets -- if not, we replace the r in the 2r by the number of useful attributes.) Hence we need $2mr^2\log_2(N / m)$ bits.
>
> Level 3: we need twice as much space as level 2 since we include the second highest frequency and the median frequency statistics too, hence $4mr^2\log_2(N / m)$ bits.
>
> Level 4: we can store a distribution either implicitly (by mathematical description) or explicitly (by listing of values). For implicit storage, we need to specify a distribution function and absolute deviations above and below it (since the original distribution is discrete, it is usually easier to use the corresponding cumulative distributions). We can use codes for common distributions (like the even distribution, the exponential, and the Gaussian), and a few additional parameters of w bits, plus the positive and negative deviation extrema of w bits each too. So space will be similar to level 3 information.
>
> But some distributions are not similar to any known distribution, and we must represent them explicitly. Storage at the level of detail of individual data items would not be cost-effective (it would necessarily take as much space as the original data), so assume data items are aggregated into approximately even groups or ranges of values (or bins) for a count distribution on the original data universe. The m-fold partitioning that defined the original sets is probably a good level of aggregation (else we would not have chosen it for the other purpose originally), so let us assume it. Then we have $m^2r^2\log_2(N / m)$ bits total. If some of the groups of values (bins) on a set are zero, we can of course omit them and save space. If fraction c of the items are zero, then we multiply the space formula by (1-c). How many such zero counts there are depends considerably on the original data distribution, the nature of the attributes, and how the m 1-sets on each attribute are defined, so we cannot give a formula.
>
> Level 5: this information is similar to level 4 except that values are associated with points of a distribution. Implicit representation by good-fit curves requires just as much space as level 4 -- we just require a fixed ordering of values along the horizontal axis instead of sorting by frequency. Explicit representation also takes the same space as for level 4 explicit representation of $m^2r^2\log_2(N / m)$, but an alternative for situations with many items not present that are present in the universe is to give pairs of values and their associated frequencies.

For all methods of this type we need storage for access structures. If user queries only involve a few named sets, we can just store the names in a separate lexicon table mapping names to unique integer identifiers, requiring $m^*r^*(l + \log_2 mr)$ bits total for the table, where l is the average length of a name, assuming al statistics on the same set are stored together.

But if users want to ask queries about arbitrary value partitions of attributes, rather than about named sets, we must also store definitions (that is, what data values belong to which set) of the sets about which we have tabulated statistics. For sets

that are partitions of numeric attributes, the upper and lower limits of the subrange are sufficient. For each such attribute we need 2mw bits. But nonnumeric attributes are more trouble, because we usually have no alternative than to list all the possible values for a set. We can code the values with a hashing function, however, and include only the codes in our lists; if there are V total nonnumeric values of nonnumeric fields, $\log_2 V$ bits are needed for the code, and we can allocate $4+\log_2 V$ bits to be safe and make hash collisions negligible. Set definition space will then be $mV(4+\log_2 V)$ for all nonnumeric attributes, assuming sets are numbered by codes consecutively and the codes do not have to be included with their definition. Thus if there are r_{num} numeric attributes of the r total, the space required is approximately

$$(2r_{num}w + V(r-r_{num})(4+\log_2 V))m$$

bits.

5.8.2. An example

Suppose we have a database of N=100,000 items, r=10 attributes, and all data items are w=16 bits long. Suppose we partition each attribute into m=100 sets for bounding purposes, giving a total of 1000 1-sets. Assume also that three of the attributes are nonnumeric, with a total of V=1000 distinct values. Then the database is 10*100,000*16 = 16,000,000 bits.

Using the formulae of the last section, level 1 information requires approximately 100*10*10=10,000 bits, level 2 information 2*100*100*10 = 200,000 bits, and level 3 information 400,000. If we can find good fits of the level 4 and 5 distributions to one of 16 standard curves parameterized by two variables, implicit specification of levels 4 and 5 will require 68 bits of information for each distribution (4 for curve type + 2*16 for parameters + 2*16 for upper and lower absolute deviations), or 1000*7*68 + 1000 * 3 * 100 * 10 = 476,000 + 3,000,000 = 3,476,000 total.

Level 4 and 5 explicit distributions require 10000*100*10 = 10,000,000 bits when there are no zero-count bins, a number approaching the size of the database. If fraction c of the bins are zero, then the storage required is 1-c times this number. But we can usually improve on this considerably with compression methods, discussed in the next section.

If all sets are named, and with names compressible to 100 bits, the symbol table requires 100*10*(100+10) = 110,000 bits. If all attributes are numeric, and all 1-sets are defined as range partitions, then set definitions will require 2*10*16*100 = 32,000 bits.

These figures are not necessarily bad, not even the level 4 and 5 explicit distributions. In many databases, storage is cheap. If a set intersection is to be compiled, or a bound is needed to determine how to perform a large join where a wrong choice may mean hours or days more time, quick reasoning with a single page fetch of precomputed statistics will be much faster than computing the actual statistic or estimating it by sampling, methods which require many page fetches. See [13] for a detailed discussion.

5.8.3. Compression techniques

A variety of compression techniques can be applied to storage of statistics, extending standard compression techniques for databases [14].

We can save space with level 4 by not specifying items with zero frequency in a set. How many fewer values this means depends on the ratio of the set sizes to the size of the universe, and the relative frequency of infrequent values. As an example, suppose the distribution of values for some attribute follows a "Zipf's Law" curve, where the frequency is the reciprocal of rank order, a curve close to the distribution of many real-world phenomena. Then a set of size n that has this distribution will contain d distinct values for that attribute, where

$$\int_{.5}^{d+.5} (C/x)dx = n \ , \quad or \quad d = .5e^{n/C} - .5$$

But we can estimate C if we know the size of the universe N and the number of distinct values D in it:

$$D = .5e^{N/C} - .5 \qquad hence \quad C = \frac{N}{\ln(2D+1)}$$

and so a general expression for the number of distinct values in random subset of the universe of size N/m is
is

$$d = .5e^{n\ln(2D+1)/N} - .5 = .5(2D+1)^{n/N} - .5$$

and we have a storage savings of d/D in level 4 versus level 5.

For level 5, we can do some compression if there are many items with zero frequency. We can represent the distribution as code-frequency pairs, where the code is a hash code for the value. Or we can list all the nonzero frequencies according to some global ordering, and keep a bit vector denoting which of all the values in the universe this list includes. For level 4, we can save space by exploiting the fact that the distribution is never increasing, and code only the successive decreases in the distribution with each point.

A related and useful compression trick is to store only the offsets from an estimate value. For instance, rather than storing the size of each set we can store a number to be added or subtracted from the "ideal" set size of the size of the universe divided by the number of sets (perhaps only sets of a particular type); or we can store a number to be added or subtracted to the "ideal" mode frequency estimate of the mode frequency of the universe times the ratio of the size of the set to the size of the universe. For example, for a set of size n on a universe set of size N, expect that the mode frequency is (n/N)M and the number of distinct values is (n/N)D, where M is the mode frequency and D the number of distinct values of the universe. Or use the more sophisticated formula for D given in the last section, if you can fit to a Zipf curve; or use that method on another curve if you can fit well to it.

Another trick is entirely omit statistics that can be estimated reasonably well, and just let the system use something derived from the estimate value in calculation. When precomputing statistics on the database for use in bounding, calculate the estimate of that statistic, using the methods described above. If it is farther than 10% (or some other criterion percentage) from the actual value, store the precomputed statistic, but otherwise do not. Then when trying to bound the size of some set later, if a needed statistic that is a type known to have been precomputed has not been stored, use the above quick estimation method to get a value; add 10% to get an upper bound on the true value, and subtract 10% to get a lower bound. This "lack-of-knowledge inference" idea can save a good deal of space when the data contains a fair number of statistically independent attributes.

6. Evaluation of the frequency-distribution bounds

We provide three approaches to evaluation of the bounds thusfar presented. First, analytic approaches for simple cases; second, inequalities among the different bounds; and third, some experiments with actual data.

6.1. A simple case

Consider the intersection of two sets when we have only level 1 information. Let the set sizes be a and b, in a universe of size N . Then we can graph the upper bound, lower bound, and an independence-assumption estimate as a function of a while holding b constant (see Figure 1). The result is a parallelogram with a base and height of length b, extending a horizontal distance N. The upper bound is the two lines defining the top of the parallelogram; the lower bound is the two lines defining the bottom; and the independence-assumption estimate, ab/N, is the diagonal.

We argue that sets users query in a typical database relation (file) will be small with respect to the full relation (file). Large sets (say more than half the size of the full relation) will tend to overlap and have significantly similar characteristics. Also, large sets take more time to process, and users can be impatient. So we can expect a and b to usually be less than N/2, and the estimate will be closer to the lower bound than the upper bound. This is because the slope of the estimate line will be (b/N)a, and the initial slope of the upper bound line a, and b/N is less than .5; and for the region of variation of $b \leqslant a \leqslant N / 2$, the estimate will stay closer to the lower bound until it is equidistant from both bounds (by symmetry of parallelograms) at a=N/2.

This suggests that if data have at most weak correlations, the level 1 upper bound is going to be farther from the true value of a set size than the lower bound. So it is not surprising we have more upper bounds on intersections than lower, because good upper bounds are more powerful at limiting possibilities. Note that more than two sets to intersect just makes this tendency stronger, since if we arbitrarily designate the largest set of three as the "third" set and compute the bounds and estimate on the other two, the inclusion of the "third" set will not decrease the upper bound, nor increase the lower bound, but will decrease the estimate as long as its size is not N.

We can prove these insights another way. An independence-assumption (multiplication of fractions) estimate with level 1 information is just the product of the set sizes n(i) divided by the size of the universe to the s-1 power, s the number of sets intersected. This estimate is always within the level 1 bounds -- that is,

$$\min_{i=1}^{n} n(i) \geqslant \frac{\prod_{i-1}^{s} n(i)}{N^{m-1}} \geqslant \sum_{i=1}^{s} n(i) - (s-1)N$$

because by taking logarithms this is equivalent to:

$$\min_{i=1}^{s} \log(n(i)) \geqslant \sum_{i=1}^{s} \log(n(i)) - (s-1)\log(N) \geqslant \log\left| \sum_{i=1}^{s} n(i) - (s-1)N \right|$$

The second inequality follows directly because $\log(x)+\log(y) \geqslant \log(x+y)$. The first inequality can be rearranged to

$$\left| \min_{i=1}^{s} \log(n(i)) \right| + (s-1)\log(N) \geqslant \sum_{i=1}^{s} \log(n(i))$$

where we can match m terms on both sides. Since for all i $\log(N) \geqslant \log(n(i))$, and since

$\min\limits_{i=1}^{s} \log(n(i)) \geqslant \log(n(j))$ **for all j, and the sum of inequalities in the same direction is an inequality in the same direction, the result follows.**

We suspect that similar phenomena hold for levels 2 through 5, but the mathematical analysis gets much more complicated.

6.2. Comparing bounds

We can prove some relationships between bounds (see Figure 2). Here are the conditions on intersection frequency-distribution bounds:

1. Level 2b upper bounds are better than level 1, since they are the sum of two positive numbers and one nonpositive number, and the first positive number is less than or equal to the mode frequency of the smallest set, and the second positive number is less than or equal to the frequency of all the rest of the items of the smallest set.

2. Level 3a upper bounds are better than level 2a upper bounds because the frequency of the second most common item and the frequency of the median item must be less than the mode frequency. Thus the same number, but smaller, terms are added with level 3 information. Hence their sum is smaller.

3. Level 3b upper bounds must be better than level 2b upper bounds by a similar argument (additional terms are being subtracted from the level 2b bound expression).

4. Level 4 upper bounds must be better than level 3a upper bounds because the set of actual frequencies is being used, not upper bounds on the frequencies (in the form of the second most common and median frequencies).

5. Level 4 upper bounds must be better than level 3b upper bounds by a similar argument.

6. Level 5 upper bounds must be better than level 4 since smaller numbers are used to calculate a bound on the kth most frequent item.

7. Level 5 lower bounds must be as good or better than level 1 lower bounds because with level 5 the sets intersected are being partitioned into many subsets based on attribute values, and a disjoint union taken which introduces no inaccuracies.

By similar lines of argument we can make exactly the same arguments for lower and upper bounds, respectively, on unions.

6.3. Experiments

There are two distinct requirements for bounds on set intersection and union sizes to be more useful than estimates of those same things:

1. Some of the sets being intersected or unioned are significantly nonindependent -- that is, not drawn randomly from some much larger population. This means that the usual estimates of their intersection size obtained from level 1 (size of the intersected sets) information will be poor.

2. At least one set being intersected or unioned has a significantly different frequency distribution from the others on at least one attribute. This can happen when at least one set has values on an attribute that are not randomly drawn.

These criteria can be justified by the general homomorphism idea behind our approach

(see section 4.2): good bounds result whenever values in the range of the homomorphism get very different counts mapped onto them for each set. These criteria can be used to decide a priori for which sets on a database it might be useful to store statistics for computing bounds. (Rigorous definition and systematic application of such criteria could support the lack-of-knowledge inferences discussed in section 5.8.3, providing further advantages.)

6.3.1. Experiments: nonrandom sets

As a simple illustration, consider the experiments summarized in the tables of Figures 3 and 4. We created a synthetic database of 300 tuples of four attributes whose values were evenly distributed random digits 0-9. We wrote a routine (MIX) to generate random subsets of the data set satisfying the above two criteria, finding groups of subsets that had unusually large numbers of items in common. We conducted 10 experiments each on random subsets of sizes 270, 180, 120, and 30. There were four parts to the experiment, each summarized in a separate table. In the top tables in Figures 3 and 4, we estimated the size of the intersection of two sets; in the lower tables. we estimated the size of the intersection of four sets. In Figure 3 the chosen sets had 95% same items; in Figure 4, 67%.

The entries in the tables represent means and standard deviations in 10 experiments of the ratios of bounds or estimates to the actual intersection size. There are four pairs of columns for the four different set sizes investigated. The rows correspond to the various frequency-distribution levels discussed: the five levels of upper bounds first, then two estimate methods, then the two lower bound methods. (Since level 5 information is just level 1 information at a finer level of detail, it is easier to generalize the level 1 estimate formula to a level 5 estimate formula.) Only level 2a and 3a rules were used, not 2b and 3b.

The advantage of bounds shows in both Figure 3 and Figure 4, but more dramatically in Figure 3 where sets have the 95% overlap. Unsurprisingly, lower bounds are most helpful for the large set sizes (left columns), whereas upper bounds are most helpful for the small set sizes (right columns). However, the lower bounds are not as useful because when they are close to the true set size (i.e. the ratio is near 1), estimates are also close. But when upper bounds are close to the true set size for small sets, both estimates and lower bounds can be far away.

6.3.2. Experiments: real data

The above experiments were with synthetic data, but we found similar phenomena with real-world data. A variety of experiments, summarized in [16], were performed with data extracted from a database of medical (rheumatology) patient records. Performance of estimate methods vs. our bounding methods was studied for different attributes, different levels of information, and different granularities of statistical summarization. Results were consistent with the preceding for a variety of set types. This should not be surprising since our two criteria given previously are often fulfilled with medical data, where different measures (tests, observations, etc.) of the sickness of a patient often tend to correlate.

7. Bounds from range analysis

We have so far examined only frequency-distribution bounds on the size of a set intersection, but as we mentioned in section 4, these are only one example of a class of bounding methods involving mappings (homomorphisms) of a set of data items onto a distribution. Another very important example are bounds obtained from analysis on the range of values for some attribute, call it A, of the data items for each set intersected. These methods essentially create new sets, defined as partitions on A, which contain the intersection being studied. These new sets can therefore can be added to the list of sets being intersected without affecting the result, and this can lead to tighter (better) bounds on the size of the intersection. There are complex ways in which these extra redundant sets can be used, but we will address only the simplest use here: as an upper bound on the size of the intersection.

7.1. Intersections on univariate ranges

We first consider reasoning about ranges of a single attribute of the data items.

7.1.1. Statistics on partitions of an attribute

All the methods (or levels) we will discuss require partition counts on some attribute A. That is, the number of data items lying in mutually exclusive and exhaustive ranges of possible values for A. For instance, we may know the number of people ages 0-9, 10-19, 20-29, etc.; or the number of people with incomes 0-9999, 10000-19999, 20000-29999, etc. We require that the attribute be sortable by something other than item frequency in order for this partioning to make sense and be different from the frequency-distribution analysis just discussed; this means that most suitable attributes are numeric.

This should not be interpreted, however, as requiring anticipation of *every* partition of an attribute that a user might mention in a query, just a covering set. To get counts on arbitrary subsets of the ranges, inequalities of the Chebyshev type may be used when moments are known, as for instance Cantelli's inequalities:

$$|\textit{probability that } x - \mu \leqslant \lambda| \leqslant \sigma^2 / \ \sigma^2 + \lambda^2$$

$$|\textit{probability that } x - \mu \leqslant \lambda| \geqslant \lambda^2 / \ \sigma^2 + \lambda^2$$

for μ the mean and σ the standard deviation of the attribute. Otherwise the count of a containing range partition may be used as an upper bound on the subset count, and a count on a contained part as a lower bound.

7.1.2. Level 1: bin counts on the universe and set ranges

Suppose we know partition (bin) counts on some numeric attribute j for the data universe. (We must know them for at least one set to apply these methods, so it might as well be the universe.) Suppose we know the maximum u(i,j) and minimum l(i,j) on attribute j for each set i being intersected. Then an upper bound on the maximum of the intersection U(j) on j is the minimum of the maxima, and a lower bound on the minimum of the intersection L(j) is the maximum of the minima, or

$$U(j) = \min_{i=1}^{s} u(i,j) \ , \ L(j) = \max_{i=1}^{s} l(i,j)$$

Note if $U(j) < L(j)$ we can immediately say the intersection is the empty set.

So the intersection set must be a subset of the set of all items in the universe that have values from L(j) to U(j) on attribute j. We can do this for any numeric attribute j. So an upper bound on the size of the intersection is the minimum-size such set over all attributes j in Q, or

$$\min_{j=1}^{r} \left(\sum_{k=B(L(j),j)}^{B(U(j),j)} f(U,j,k) \right)$$

where s sets are intersected; where there are r numeric attributes; where B(x,j) denotes the number of the bin into which value x falls on attribute j; and where f(U,j,k) is the number of items in partition (bin) k on attribute j for the universe U.

As an example, suppose we are intersecting two sets whose ranges on some attribute j are 100-482 and 361-1255 respectively. Suppose we have tabulated the number of items in every range of 50 of attribute j, so we know how many items are 200-249, 250-299, etc. Then an upper bound on the size of the intersection is the sum of the sizes of the sets defined by the ranges 350-399, 400-449, and 450-499.

7.1.3. Extensions of level 1

The above approach can be improved with more knowledge of the sets intersected. If the requirement of maxima and minima for all the sets is too difficult, you can store only the maxima and minima most different from the corresponding maxima and minima for the universe, and just ignore the nonstored statistics in your computation. You also can save half the space by using ranges of the sets (the difference between the maximum and the minimum for each set), and an upper bound on the range of the intersection is the minimum of the ranges. Then the indices of k must run over the adjacent bins (whose number is determined by the range) with the largest total count.

Absolute bounds on correlations between attributes may also be exploited. If two numeric attributes have a strong relationship to each other, we can formally characterize a mapping from one to the other with three items of information: the algebraic formula, an upper deviation from the fit to that formula, and a lower deviation. We can calculate these three things for pairs of numeric attributes on the universe set, and store only the information for pairs with strong correlations. To use correlations in finding upper bounds, we just substitute a more complicated method of finding L(j) and U(j) in the above method. First, for every attribute j we find L(j) and U(j) by the old method, i.e.

$$U(j) = \min_{i=1}^{s} u(i,j) \;,\; L(j) = \max_{i=1}^{s} l(i,j)$$

Then, for every stored correlation from an arbitrary attribute c to an arbitrary attribute d. we calculate the projection of the range of c (from L(c) to U(c)) by the formula onto d. The overlap of this range on the original range of d (from L(d) to U(d)) is then the new range on d and L(d) and U(d) are updated if necessary. Applying these correlations requires relaxation methods since narrowing of the range of one attribute may allow new and tighter narrowings of ranges of attributes to which that attribute correlates, which may entail further narrowings of the original attribute, and so on.

Functional dependencies can also be used. (Also what we have termed elsewhere [11] extensional functional dependencies", true only for the current database state, and handle mappings not functional but relational with a fixed maximum number of image values for a given domain value.) Suppose you only go as far as calculating the f(U,j,k) values in the method of the last section. Now if there is a functional dependency from some attribute j to some attribute i, and the number of distinct values on this subrange of j is d, then the number of distinct values on the determined subrange of i must be at most d. If you know how many distinct values of i are in each bin, this then bounds the number of bins on i, and hence the size of the intersection set you are analyzing.

7.1.4. Level 2: mode frequencies on bin counts for intersected sets

At the next level of information, analogous to level 2 for frequency-distribution bounds, we have information about distributions of values for particular attributes. Suppose this includes an upper bound m(i,j) on the number of things in set i in a bin, of some attribute j. (This m(i,j) is like the mode frequency in section 5, except the equivalence class here is all items in a certain range on a certain attribute.) Assume also we know how many bins a given range of an attribute covers. Then the formula for an upper bound on the size of the set intersection is

$$\min_{j=1}^{r}\left((B(U(j),j)-B(L(j),j)+1)*(\min_{i=1}^{s} m(i,j)))\right)$$

where U(j) and L(j) are as before.

As an example, take the situation in the example before where we conclude that an upper bound on the size of the intersection of two sets is the sum of the counts on the ranges 350-399, 400-449, and 450-499 on some attribute j. If we know that the upper bounds on the bin counts on j for the two sets are 28 and 33, then an upper bound on the size of the intersection is 3*28 = 84.

7.1.5. Level 5: bin counts for intersected sets

Finally, if we know the actual distribution of bin counts for each set i being intersected, we can modify the formula of level 1 as follows:

$$\min_{j=1}^{r}\left(\sum_{k=B(L(j),j)}^{B(U(j),j)} \left[\min_{i=1}^{s} f(i,j,k)\right]\right)$$

where s sets are intersected; where there are r numeric attributes; where B(x,j) denotes the number of the bin into which value x falls on attribute j; and where f(i,j,k) is the number of items in partition (bin) k on attribute j for set i.

As with frequency-distribution level 4 and level 5 bounds, we can also use this formula when all we know is an upper bound on the bin counts, perhaps from a verbal description of an upper bound curve like "normal with mean 10.6 and standard error 2.3" -- we just get a weaker result.

7.2. Multidimensional intersection range analysis

We now consider range analysis for multidimensional ranges. That is, the projection of data items onto some multidimensional space representing values for some subset S of attributes.

7.2.1. Level 1: Set ranges and bounds on the universe

Analogous to Level 1 univariate range analysis, we may be able to give a multivariate distribution that is an upper bound on the distribution of the universe set over S. We determine ranges on each attribute of S by finding the overlap of the ranges for each set being intersected as before. This defines a hyperrectangular region in hyperspace, and the universe upper bound puts an upper bound on the number of items inside it. We can also use various multivariate generalizations of Chebyshev's inequality [1] to bound the number of items in the region from knowledge of moments of any set containing the intersection set (including the universe).

7.2.2. Extensions of level 1

As with univariate range analysis, we can exploit known correlations to further truncate the ranges on each attribute of S, obtaining a smaller hyperrectangular region. Functional dependencies can also be used as before.

Another class of correlation we can use is specific to multivariate ranges: those between attributes in the set S itself. For instance, a tight linear correlation between two numeric attributes a and b, strongly limits the number of items within rectangles the regression line does not pass through. If we know absolute bounds on the regression fit we can infer zero items within whole subregions. If we know a standard error on the regression fit we can use Chebyshev's inequality and its relatives to bound how many items can lie certain distances from the regression line.

7.2.3. Levels 2 and 5

Just as for univariate range analysis, we can exploit more detailed information about the distributions of any attribute (not necessarily the ones in S). If we know an upper bound on bin size, for some partitioning into subregions or "bins", or if we know the exact distribution of bin sizes, we can improve on the level 1 bounds.

7.3. Range analysis of unions

Formulae for unions are straightforward to obtain. For level 1 information we just change our calculation for the upper and lower bounds in the summation, substituting the maximum over the sets for the minimum, and the minimum over the sets for the maximum. For level 2 and level 5 information, in addition to the bounds change we must substitute a summation for the minimum over i, analogous to the situation for frequency-distribution bounds.

7.4. Embedded query expressions for range analysis

We consider only level 1 information because the other levels are too hard to analyze. We cannot handle query expressions with set complements in them, because there is no good way to determine a maximum or minimum of the complement of a set other than the maximum or minimum of the universe. Also we cannot handle query expressions with intersections inside unions and vice versa, since our range analysis provide only upper bounds on intersections and unions.

Statistics on the universe are used with the level 1 calculation, and the only set-dependent information are the extrema of the range, U and L. Equivalence of query expressions under commutativity or associativity of terms in intersections or unions then follows from the commutivity of the maxima and minima of operations. Equivalence under reflexivity follows because the maximum or minimum of multiple occurrences of the same number is that number itself. Similarly, occurrences of the universe set and the null set are useless, because the maximum of any nonnegative number and zero is always the number, and the minimum of a number less than the size of the universe and the size of the universe is the number. So query rearrangements of embedded queries for range analysis bounds cannot improve the bounds and we might as well not bother.

7.5. Storage requirements for range analysis

Space requirements for these range analysis bounds can be computed in the same way as for the frequency-distribution bounds. Assume that the number of bins on each attribute is m, the number of attributes is r, the number of bits required for each attribute value is w, and the number of items in the database is N. Then the space requirements for univariate range bounds are:

$$\textit{level 1: } mr\log_2(N/m)+2mr^2w$$

$$\textit{level 2: } 2mr^2w+mr^2\log_2(N/m)$$

$$\textit{level 5: } 2mr^2w+m^2r^2\log_2(N/m)$$

Again, these are pessimistic estimates since they assume that all attributes can be helpful for range analysis. In many databases this is not the case, and these formulae can be reduced proportionately by the fraction of the attributes that show significantly different distributions for different sets.

7.6. Evaluation of the range-analysis bounds

The performance of the range-analysis bounds is much harder to predict than the frequency-distribution bounds since it depends much more on the nature of the data distributions. Thus we do not analyze it here.

7.7. Cascading range-analysis and frequency-distribution methods

The above determination of the maximum and minimum of an intersection set on an attribute can be used to find better frequency-distribution bounds too, since it effectively adds new sets to the list of sets being intersected, sets defined as partitions of the values of particular attributes. These new sets may have unusual distributions on further attributes that can lead to tight frequency-distribution bounds.

8. Handling updates to the database

A consideration besides accuracy for what information to precompute towards helpful bounds is database updates. We can represent a list of updates as a list of data item insertions and deletions. Then keeping level 1 and level 5 information correct (which is just counts on various sets) is easy: we add one to a set's count for every insertion to that set, and subtract one for every deletion. Thus no recomputation of the count on the database is needed, just a single lookup for each item inserted or deleted to determine what sets it belongs to.

The number of unique items in a set can also be updated the same way. And any moment statistic (including means and standard deviations) can be updated without going to the database, provided the corresponding set count is also known. We can keep maxima and minima up to date with insertions (just take the more extreme of the old value and the insertion value), and most of the time with deletions too, because for large sets with random values being deleted, a deletion of the extremum is not very likely. Note the statistics mentioned so far in level 1 and level 5 for both frequency-distribution and range-analysis bounds, so those bounds are easy to keep up to date.

The other kinds of precomputed statistics mentioned here -- mode frequencies, median frequencies, best-fit distributions -- are trickier to update, and sometimes we have no choice but to go to the database to recompute them. (There are databases with few or no updates for which these statistics can be used without qualms, such as most statistical databases [15].) But note if we know the mode, the mode frequency m(i,j), and the frequency of the second most common item m2(i,j), the mode frequency can be updated without trouble provided the difference of the number of inserts and the number of deletes for any other item is no more than m(i,j)-m2(i,j) more than the difference of the number of inserts and the number of deletes to the mode. Some similar rules can be given for the other frequency statistics.

9. Conclusion

We have presented a comprehensive approach to bounding sizes of intersections, unions, and complements of sets in a database, providing a library a formulae for different situations. We have emphasized intersections (because of their greater importance) and upper bounds (because they are often easier to obtain). Our methods exploit simple precomputed statistics (counts, frequencies, maxima and minima, and distribution fits) on general-purpose sets in the database. The more we precompute, the better our bounds can be. We illustrate by analysis and experiments the time-space-accuracy tradeoffs involved, guiding selection of the best bound formula in a given situation. Our bounds tend to be most useful when there are strong or complex correlations between sets mentioned in a query, a situation in which estimation methods for set size tend to do poorly. This work thus nicely complements those methods.

References

Author's address: Neil C. Rowe, Department of Computer Science, Code 52, Naval Postgraduate School, Monterey, CA 93943 USA.

Acknowledgements: The work reported herein was partially supported by the Foundation Research Program of the Naval Postgraduate School with funds provided by the Chief of Naval Research. The views and conclusions contained in this document are those of the author and should not be interpreted as representative of the official policies of the Navy or the U.S. Government. Barry Tilden provided helpful discussion. Judy Quesenberry helped with typing.

[1] H. W. Block and A. R. Sampson, "Inequalities on distributions: bivariate and multivariate," in *The Encyclopedia of Statistical Sciences*, volume 4, New York: Wiley, 1983, 76-82.

[2] S. Christodoulakis, "Estimating record selectivities," *Information Systems, 8*, 2, 1983, 105-115.

[3] L. H. Cox, "Suppression methodology and statistical disclosure control," *Journal of the American Statistical Association, 75*, 370, June 1980, 377-385.

[4] R. Demolombe, "Estimation of the number of tuples satisfying a query expressed in predicate calculus language," Proceedings of the Sixth Conference on Very Large Data Bases, September 1980, 55-63.

[5] D. E. Denning and J. Schlorer, "Inference controls for statistical databases," *IEEE Computer, 16*, 7, July 1983, 69-81.

[6] E. L. Lawler, "An approach to multilevel boolean minimization," *Journal of the ACM, 11*, 3, July 1964, 283-295.

[7] E. Lefons, A. Silvestri, and F. Tangorra, "An analytic approach to statistical databases," Proceedings of the Ninth International Conference on Very Large Data Bases, Florence, Italy, September 19833, 260-274.

[8] W. Lipski, "On semantic issues connected with incomplete information databases," *ACM Transaction on Database Systems, 4*, 3, September 1979, 262-296.

[9] G. Piatetsky-Shapiro and C. Connell, "Accurate estimation of the number of tuples satisfying a condition," Proceedings of the ACM-SIGMOD Annual Meeting, Boston, Mass., June 1984, 256-276.

[10] P. Richard, "Evaluation of the size of a query expressed in relational algebra," Proceedings of the ACM-SIGMOD Annual Meeting, June 1981, 155-163.

[11] N. C. Rowe, "Rule-based statistical calculation on a database abstract," Report STAN-CS-83-975, Stanford University Computer Science Department, June 1983 (Ph. D. thesis).

[12] N. C. Rowe, "Diophantine inferences on a statistical database," *Information Processing Letters, 18*, 1984, 25-31.

[13] N. C. Rowe, "Antisampling for estimation: an overview," *IEEE Transactions on*

Software Engineering, to appear October 1985.

[14] D. Severance, "A practitioner's guide to data base compression," *Information Systems, 8,* 1, 1983, 51-62.

[15] A. Shoshani, "Statistical databases: characteristics, problems, and some solutions," Proceedings of the 8th International Conference on Very Large Data Bases, Mexico City, Mexico, 1982, 208-222.

[16] B. M. Tilden, "A hierarchy of knowledge levels implemented in a rule-based production system to calculate bounds on the size of intersection and unions of simple sets," M. S. Thesis, U. S. Naval Postgraduate School, December 1984.

[17] J. D. Tukey, *Exploratory Data Analysis,* Reading, Mass.: Addison-Wesley, 1977.

Appendix: Best equivalent forms for level 1 frequency-distribution bounds

We give here the detailed comparison of level 1 frequency-distribution bounds (both upper and lower) for query set expressions equivalent under Boolean algebra. We first summarize the six level 1 bounds we have obtained in sections 5.1.1, 5.2.1, 5.4, and 5.5.2. Here "sup" denotes upper bound, and "inf" lower bound. A(i) denotes the ith set, n(i) denotes its size, s the number of sets intersected, and N the size of the data universe.

$$sup(n\left(\bigcap_{i=1}^{s} A(i)\right)) = \min_{i=1}^{s} n(i)$$

$$inf(n\left(\bigcap_{i=1}^{s} A(i)\right)) = \max(0, \left(\sum_{i=1}^{s} n(i)\right) - (s-1)N)$$

$$sup(n\left(\bigcup_{i=1}^{s} A(i)\right)) = \min(N, \sum_{i=1}^{s} n(i))$$

$$inf(n\left(\bigcup_{i=1}^{s} A(i)\right)) = \max_{i=1}^{s} n(i)$$

$$sup(n(\bar{A}(i))) = N - inf(n(A(i)))$$

$$inf(n(\bar{A}(i))) = N - sup(n(A(i)))$$

9.1. Commutativity

Given s sets to find the intersection or union of, the order in which the sets are specified does not matter because examination of the rules shows this only changes the order of a sum, minimum, or maximum, and those operations are commutative.

9.2. Reflexivity

Since

$$sup(n(A \cap A)) = \min(a, a) = a \;,\; inf(n(A \cap A)) = \max(0, 2a - N)$$

$$sup(n(A \cup A)) = \min(N, 2a) \;,\; inf(n(A \cup A)) = \max(a, a) = a$$

the bounds will not be the same unless a=N in the first case or a=0 in the second. Hence the equivalent expression of just the set A is preferable for obtaining bounds.

9.3. Associativity of intersection

The next question is whether associative grouping of some of the sets in an intersection or union might affect the result. Let

$$Q1 = \bigcap_{i=1}^{s} A(i) \;,\; Q2 = \left(\bigcap_{i=1}^{k} A(i)\right) \cap \left(\bigcap_{k+1}^{s} A(i)\right)$$

where k is some arbitrary integer between 1 and s. (By embedding these groupings, we can model an arbitrary associative computation scheme.) Then for upper bounds:

$$sup(n(Q1)) = \min_{i=1}^{s} n(i) \;,\; sup(n(Q2)) = \min(\min_{i=1}^{k} n(i) \;,\; \min_{i=k+1}^{s} n(i))$$

and since the min operator is associative, the bounds are equivalent.

Similarly, we can show that the lower bounds are equivalent, though it is more difficult. The bounds are:

$$inf(n(Q1)) = \max(0, \left(\sum_{i=1}^{s} n(i)\right) - (s-1)N)$$

$$inf(n(Q2)) = \max(0, \max(0, \left(\sum_{i=1}^{k} n(i)\right) - (k-1)N) + \max(0, \left(\sum_{i=k+1}^{s} n(i)\right) - (s-k-1)N) - N)$$

We have three cases to consider for each of the inner max expressions for Q2:

(1) Suppose the second argument of both is the larger; then the lower bound expression for Q2 becomes the same as that for Q1.

(2) Second, suppose the first argument (0) is larger for the first inner max expression. (This includes the case where the first argument of the second inner max is larger at the same time.) Since

$$\left(\sum_{i=k+1}^{s} n(i)\right) - (s-k-1)N \leqslant N$$

the outer max must be zero. So the Q2 lower bound is zero. But since

$$[\left(\sum_{i=1}^{s} n(i)\right) - (s-1)N] = -N + [\left(\sum_{i=1}^{k} n(i)\right) - (k-1)N] + [\left(\sum_{i=k+1}^{s} n(i)\right) - (s-k-1)N]$$

and we have assumed the second term in brackets is less than 0, and we have shown that the third term in brackets is less than or equal to N, the first term in brackets must be less than 0. Hence the Q1 lower bound is zero too.

(3) Third, suppose the first argument (0) of the second inner max in the Q2 bound is larger. Then by analogous reasoning to the preceding, the Q1 and Q2 bounds are equal.

9.4. Associativity of union

From the last section it follows that associativity does not matter to set unions, because any union of s sets can be written as the complement of the intersection of the complements of those sets, and there is no additional uncertainty introduced in the handling of complements of sets (just subtract the size or the bound from N).

9.5. Distributivity of intersection over union

But it turns out that the distributive laws of intersection over union, and union over intersection, do not preserve bounds: the factored form is preferable. To see this, first consider distribution of intersection over union:

$$Q3 = A \cap \left(\bigcup_{i=1}^{s} B(i)\right), \quad Q4 = \bigcup_{i=1}^{s} (A \cap B(i))$$

Then the upper bounds on the sizes of Q3 and Q4 are:.

$$sup(n(Q3)) = \min(a, \min(N, \sum_{i=1}^{s} b(i))) = \min(a, \sum_{i=1}^{s} b(i))$$

$$sup(n\ (Q\ 4)) = \min(N, \sum_{i=1}^{s} \min(a, b\ (i)))$$

Assume case 1: $b\ (i) \geq a$ for some i. Then $a = \min(a, b\ (i))$ for some i, and since a and b(i) are always nonnegative, $a \leq \min(a, b\ (i))$. Assume case 2: $b\ (i) < a$ for all i. Then $b\ (i) = \min(a, b\ (i))$ and

$$\sum_{i=1}^{s} b\ (i) = \sum_{i=1}^{s} \min(a, b\ (i))$$

Hence under all circumstances

$$\sum_{i=1}^{s} b\ (i) \leq \sum_{i=1}^{s} \min(a, b\ (i)) \text{ and } a \leq N$$

and so the upper bound on Q3 (the "factored out" form) is always less than or equal to the upper bound on Q4, and hence preferable.

The lower bounds on Q3 and Q4 are:

$$sup(n\ (Q\ 3)) = \max(0, a - N + \max_{i=1}^{s}\ b\ (i))$$

$$inf(n\ (Q\ 4)) = \max_{i=1}^{s}(\max(0, a - N + b\ (i)))$$

But the two bounds are equivalent, since

$$\max_{i=1}^{s}(\max(0, a - N + b\ (i))) = \max(0, \max_{i=1}^{s}(a - N + b\ (i))) = \max(0, a - N + \max_{i=1}^{s}\ b\ (i))$$

since a and N do not vary with i.

9.6. Distributivity of union over intersection

Similar analysis shows that the factored form for distribution of union over intersection is also preferable. Let

$$Q\ 5 = A \bigcup \left(\bigcap_{i=1}^{s} B\ (i) \right),\quad Q\ 6 = \bigcap_{i=1}^{s} (A \bigcup B\ (i))$$

Then the upper bounds are:

$$sup(n\ (Q\ 5)) = \min(N, a + \min_{i=1}^{s}\ b\ (i))$$

$$sup(n\ (Q\ 6)) = \min_{i=1}^{s}(\min(N, a + b\ (i)))$$

But

$$\min_{i=1}^{s}(\min(N, a + b\ (i))) = \min(N, \min_{i=1}^{s}(a + b\ (i))) = \min(N, a + \min_{i=1}^{s}\ b\ (i))$$

and the upper bounds on Q5 and Q6 are equivalent.

The lower bounds are:

$$inf(n\ (Q\ 5)) = \max(a, \max(0, \sum_{i=1}^{s}(b\ (i)) - (s-1)N)) = \max(a, \sum_{i=1}^{s}(b\ (i)) - (s-1)N)$$

$$inf(n\ (Q\ 6)) = \max(0, \sum_{i=1}^{s} \max(a, b\ (i)) - (s-1)N)$$

For case 1, assume that $b(i) > a$ for all i. Then $\max(a, b(i)) = b(i)$ for all i, and the two bounds are equivalent. For case 2, assume that $b(j) \leqslant a$ for some j. Then since $0 \geqslant \max(a, (b(i)) - N$:

$$0 \geqslant \left(\sum_{i=1, j \neq i1}^{s} \max(a, b(i)) \right) - (s-1)N$$

$$a \geqslant \left(\sum_{i=1}^{s} \max(a, b(i)) \right) - (s-1)N$$

Hence the Q5 bounds is always greater than or equal to the Q6 bound, and again the factored form is preferable.

9.7. The universal set, the null set, and absorption

Let U represent the universe set and Φ the empty set. Then:

$$sup(n(A \bigcup U)) = \min(N, a+N) = N \ , \ inf(n(A \bigcup U)) = \max(a, N) = N$$

$$sup(n(A \bigcap U)) = \min(a, N) = a \ , \ inf(n(A \bigcap U)) = \max(0, a+N-N) = a$$

$$sup(n(A \bigcup \Phi)) = \min(N, a+0) = a \ , \ inf(n(A \bigcup \Phi)) = \max(a, 0) = a$$

$$sup(n(A \bigcap \Phi)) = \min(a, 0) = 0 \ , \ inf(n(A \bigcap \Phi)) = \max(0, a+0-N) = 0$$

So it does not matter to the bounds whether we take one of the above forms or simply the equivalent expression of the single set A.

The above can be applied to the "absorption" laws:

$$A \bigcap (A \bigcup B) = (A \bigcup \Phi) \bigcap (A \bigcup B) = A \bigcap (\Phi \bigcup B) = A \bigcap B$$

$$A \bigcup (A \bigcap B) = (A \bigcap U) \bigcup (A \bigcap B) = A \bigcup (U \bigcap B) = A \bigcup B$$

so since factoring is preferable, the last "absorbed" forms are preferable.

9.8. Negation equivalences

We have not yet considered negation, but it causes few difficulties. First note

$$sup(n(A \bigcap \bar{A})) = \min(a, N-a) \ , \ inf(n(A \bigcap \bar{A})) = \max(0, a+N-a-N) = 0$$

$$sup(n(A \bigcup \bar{A})) = \min(N, a+N-a) = N \ , \ inf(n(A \bigcup \bar{A})) = \max(a, N-a)$$

so better bounds are usually obtained by replacing $A \bigcup \bar{A}$ with U, and $A \bigcap \bar{A}$ with Φ.

We can use this to show another form of absorption is desirable:

$$A \bigcap (\bar{A} \bigcup B) = (A \bigcap \bar{A}) \bigcup (A \bigcap B) = \Phi \bigcup (A \bigcap B) = A \bigcap B$$

$$A \bigcup (\bar{A} \bigcap B) = (A \bigcup \bar{A}) \bigcap (A \bigcup B) = U \bigcap (A \bigcup B) = A \bigcup B$$

We must also consider DeMorgan's Laws:

$$sup(n(\overline{A \bigcap B})) = N - \max(0, a+b-N) \ , \ sup(n(\bar{A} \bigcup \bar{B})) = \min(N, (N-a)+(N-b))$$

But

$$\min(N, (N-a)+(N-b)) = N + \min(0, N-a-b) = N - \max(0, a+b-N)$$

so the two bounds are equivalent. Similar reasoning can show that the following analogous cases have equivalent bounds:

$$inf(n(\overline{A \cap B})) = N - \min(a,b)\ ,\ inf(n(\bar{A} \cup \bar{B})) = \max(N-a, N-b)$$

$$sup(n(\overline{A \cup B})) = N - \max(a,b)\ ,\ sup(n(\bar{A} \cap \bar{B})) = \min(N-a, N-b)$$

$$inf(n(\overline{A \cup B})) = N - \min(N, a+b)\ ,\ inf(n(\bar{A} \cap \bar{B})) = \max(0, a+b-N)$$

Captions for figures

Figure 1: estimates and level 1 bounds on the size of the intersection of two sets size a and b, as a function of a, in a universe of size N

Figure 2: strength relationships between the frequency-distribution bounds on intersections

Figure 3: experiments measuring average ratio of bounds and estimates to actual intersection size, 95% set overlap (95% of the items in each set are in the other(s)). Entries give ratio followed by standard error.

Figure 4: experiments measuring average ratio of bounds and estimates to actual intersection size, 67% set overlap (67% of the items in each set are also in the other(s)). Entries give ratio followed by standard error.

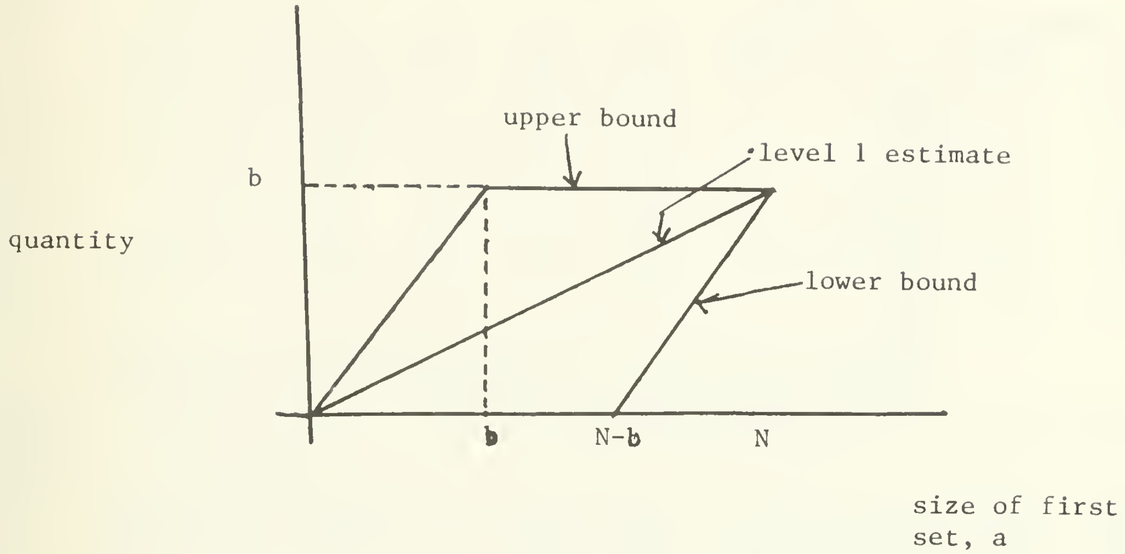
upper bound
level 1 estimate
b
quantity
lower bound
b
N-b
N
size of first set, a

Figure 1

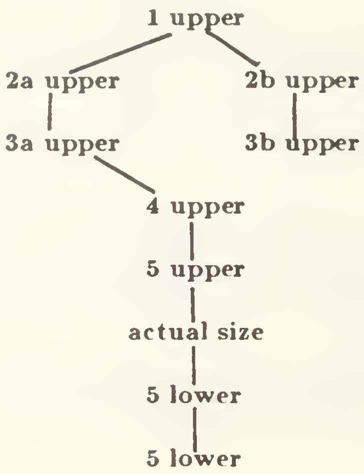
1 upper
2a upper
2b upper
3a upper
3b upper
4 upper
5 upper
actual size
5 lower
5 lower

Figure 2

Average ratio of bounds and estimates to actual intersection size for two sets chosen by the MIX routine to have 95% overlap

number of sets	kind of bound or estimate	set size 270		set size 180		set size 120		set size 30	
2	level 1 upper bound	1.05	0.0	1.05	0.0	1.04	0.0	1.03	0.0
2	level 2a upper bound	1.28	0.01	1.31	0.05	1.37	0.03	1.63	0.08
2	level 3a upper bound	1.11	0.01	1.12	0.01	1.12	0.02	1.15	0.03
2	level 4 upper bound	1.02	0.0	1.02	0.0	1.01	0.01	1.0	0.0
2	level 5 upper bound	1.02	0.01	1.01	0.01	1.01	0.01	1.0	0.0
2	level 1 estimate	0.94	0.0	0.63	0.0	0.42	0.0	0.1	0.0
2	level 5 estimate	0.95	0.0	0.64	0.0	0.43	0.0	0.13	0.0
2	level 5 lower bound	0.93	0.0	0.42	0.01	0.07	0.01	0.0	0.0
2	level 1 lower bound	0.93	0.0	0.35	0.0	0.0	0.0	0.0	0.0

Average ratio of bounds and estimates to actual intersection size for four sets chosen by the MIX routine to have 95% overlap

number of sets	kind of bound or estimate	set size 270		set size 180		set size 120		set size 30	
4	level 1 upper bound	1.15	0.01	1.13	0.01	1.13	0.01	1.1	0.02
4	level 2a upper bound	1.39	0.02	1.43	0.06	1.45	0.05	1.65	0.02
4	level 3a upper bound	1.2	0.01	1.19	0.03	1.2	0.02	1.19	0.04
4	level 4 upper bound	1.1	0.01	1.08	0.01	1.08	0.01	1.03	0.01
4	level 5 upper bound	1.08	0.01	1.06	0.01	1.06	0.01	1.01	0.01
4	level 1 estimate	0.83	0.01	0.24	0.0	0.07	0.0	0.0	0.0
4	level 5 estimate	0.85	0.01	0.28	0.0	0.09	0.0	0.0	0.0
4	level 5 lower bound	0.76	0.01	0.04	0.01	0.0	0.0	0.0	0.0
4	level 1 lower bound	0.76	0.01	0.0	0.0	0.0	0.0	0.0	0.0

Figure 3

Average ratio of bounds and estimates to actual intersection size for two sets chosen by the MIX routine to have 67% overlap

number of sets	kind of bound or estimate	set size 270		set size 180		set size 120		set size 30	
2	level 1 upper bound	1.11	0.01	1.49	0.05	1.5	0.04	1.44	0.09
2	level 2a upper bound	1.35	0.04	1.87	0.14	1.94	0.11	2.02	0.22
2	level 3a upper bound	1.17	0.01	1.53	0.06	1.51	0.06	1.43	0.12
2	level 4 upper bound	1.08	0.01	1.38	0.06	1.32	0.02	1.29	0.08
2	level 5 upper bound	1.05	0.01	1.29	0.05	1.25	0.04	1.13	0.05
2	level 1 estimate	1.0	0.0	0.89	0.03	0.6	0.02	0.14	0.01
2	level 5 estimate	1.0	0.0	0.9	0.03	0.62	0.02	0.16	0.01
2	level 5 lower bound	0.99	0.0	0.55	0.01	0.1	0.02	0.0	0.0
2	level 1 lower bound	0.99	0.0	0.5	0.02	0.0	0.0	0.0	0.0

Average ratios of bounds and estimates to actual intersection sizes for two sets chosen by the MIX routine to have 67% overlap

number of sets	kind of bound or estimate	set size 270		set size 180		set size 120		set size 30	
4	level 1 upper bound	1.34	0.03	2.94	0.17	2.86	0.26	3.07	0.35
4	level 2a upper bound	1.6	0.04	3.66	0.25	3.74	0.37	4.01	0.76
4	level 3a upper bound	1.38	0.03	2.99	0.17	2.93	0.29	2.94	0.36
4	level 4 upper bound	1.27	0.02	2.61	0.14	2.5	0.22	2.51	0.32
4	level 5 upper bound	1.23	0.04	2.38	0.14	2.27	0.19	1.98	0.2
4	level 1 estimate	0.98	0.02	0.64	0.04	0.18	0.02	0.0	0.0
4	level 5 estimate	0.98	0.02	0.69	0.03	0.24	0.02	0.01	0.0
4	level 5 lower bound	0.89	0.02	0.14	0.06	0.0	0.0	0.0	0.0
4	level 1 lower bound	0.89	0.02	0.0	0.0	0.0	0.0	0.0	0.0

Figure 4

INITIAL DISTRIBUTION LIST

Defense Technical Information Center 2
Cameron Station
Alexandria, VA 22314

Dudley Knox Library 3
Code 0142
Naval Postgraduate School
Monterey, CA 93943

Office of Research Administration 1
Code 012
Naval Postgraduate School
Monterey, CA 93943

Chairman, Code 52Hq 60
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943

Associate Professor Neil C Rowe, Code 52Rp 25
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943

Dr. Robert Grafton 1
Code 433
Office of Naval Research
800 N. Quincy
Arlington, VA 22217

Dr David W. Mizell 1
Office of Naval Researcn
1030 East Green Street
Pasadena, CA 91106